AMENDMENTS TO THE SPECIFICATION

Please amend the paragraph at lines 4-6 on page 1, under the subheading "Related Applications", as follows:

-- This application ~~is a continuing application of~~claims priority to commonly-owned and co-pending U.S. Provisional Application Nos. 60/141,208, filed June 25, 1999, and 60/188,659, filed March 10, 2000, each of which is hereby incorporated by reference. --

Please replace the paragraph beginning on page 6, line 14 and ending on page 7, line 4, as follows:

-- The speed of the processors used in a cluster of the invention has a direct bearing on the overall speed of the cluster. Likewise, the number of processors used in a cluster is also a factor in determining the overall speed of the cluster. The third factor that has a bearing on cluster speed is known as "architectural overhead." The architecture of a cluster not only includes the physical topology of the system, but also the software technology employed in the connectivity, which dictates how efficiently data is being distributed between processors in a cluster. Connectivity is typically not one hundred percent efficient, and it is the overhead associated with connectivity that slows the cluster. The prior art used in a complex router, plus a group of partial differential equations, to distribute ~~internodel~~internodal data. This complexity of the prior art ~~means~~indicates that the addition of one processor provided less than one processor's worth of computer power. Further, in the prior art, the efficiency of adding additional processors decreased with the number of processors added. The invention described herein uses the logical topology of a switch-interconnected Howard Cascade to ensure that free physical lines are always available. Using a linear load factor of approximately 1.2 is a reasonable overhead estimate associated with the invention. This factor is quite efficient when compared to prior art allocation methods, and is the result of using a mathematical cascade in allocating the problem sets among nodes for processing and defining internode timing. --

Please replace the paragraph spanning page 15, line 37 through page 17, line as follows:

-- Because the data distribution problem becomes more difficult when the nodes (processors or threads) are passing information amongst themselves, the following description provides for how this may be handled. There are four known methods of passing information: one to one, one to many, many to many, and many to one. The one to one method of information passing requires a single pair

of nodes known as the source and destination nodes. In order to pass information effectively both the source and destination nodes need to be time synchronized. As can be seen in **FIGs. 3A** and **3B** those nodes shown at time step 2 are time synchronized as well as those nodes shown at time step 3. Further, each of these nodes is guaranteed to have a clear channel. Thus as long as the pairs are time aligned via the Howard Cascade, point to point data transfer occurs without waiting and with clear *a priori* node pairing locations. The one to many information passing method works similarly to the one to one information transfer method. As long as the associated time synchronized nodes are used in the one to many information passing then we are guaranteed to have clear channels among all of the nodes. Because the invention preferably use TCP/IP for ~~internodel~~internodal communication, for one to many information transfer the IP broadcast and multicast options are used to send data in parallel to the proper destinations. Many to many information transfer performs like several one to many information transfers. Many to one information transfers require different processing depending on why the many to one relationship exists. For example: representing a multiple precision value of a transcendental function requires that function to be represented as a series where components of the series are added/subtracted to produce a single multiple precession instance of the function. Since all components of the series are summed, this can be thought of as a many to one relationship. However, the summing can take place as several discrete one to one relationships, thus any many to one relationship which can be represented as a discrete one to one relationships will be. Almost all current mathematical functions which are used in parallel processing fall into this category. Those few that do not will not run efficiently on any parallel processing system. The above description ~~means~~indicates that the invention is able to handle mathematical functions used in parallel processing more efficiently than the existing art. Those skilled in the art should appreciate that other clusters, cascades and supercomputing --

Please amend the paragraph at lines 10-12 on page 20 as follows:

-- The combination of the rapid allocation, automatic bandwidth matching, and efficient problem set distribution ~~means~~signifies that machines of different types and speeds can be used effectively within the cluster, providing a significant advantage of the invention. --

Please amend the paragraph at lines 18-25 on page 21 as follows:

-- **FIG. 4C** illustrates the relationship between cluster "strips" and threads. The number of threads in a "strip" equals the associated cascade level less one. For example, the number of threads in $Strip_1$ is 3. This is equivalent to cascade level 2. The number of threads in $Strip_2$ is 1, which is equivalent to cascade level 1. The number of threads in $Strip_3$ is 0, which is equivalent to cascade level 0. To cover a full strip's worth of data, the first strip needs to have addresses for the other nodes in a cascade. Given a cascade of level X, a strip at X_N ~~means~~signifies that $y = c\,(2^{N-1}-1)$, where N is the cascade level equivalent for a strip. --

Please replace the paragraph beginning.on page 24, line 28 and ending on page 25, line 9, as follows:

-- The invention provides many processing steps. These steps are explicitly set forth in ~~the provisional applications~~U.S. Provisional Application Nos. 60/141,208, filed June 25, 1999, and 60/188,659, filed March 10, 2000, and specifically within source code therein, from which this application claims priority. Nevertheless, **FIGs. 5, 5A, 5B, 5C, 5D, 5E, 5F, 5G** illustrate certain flowcharts illustrating application of the invention from an end user computer (e.g., a computer at customer entity 22, **FIG. 1**), to the home node , and to the cluster (e.g., such as the Cluster 26, **FIG. 1**). Specifically, **FIG. 5** illustrates process methodology occurring at the end user computer, **FIG. 5A** illustrates process methodology occurring at the home note, and **FIG. 5B** illustrates process methodology occurring at the cluster. **FIG. 5C, FIG. 5D, FIG. 5E, FIG. 5F, FIG. 5G** illustrate library actions and interconnections (e.g., connectors A-F) with **FIG. 5**, FIG. 5A, **FIG. 5B** and other flowcharts of **FIGs. 5C-5G**. As used within these flowcharts: 'a Howard Strip" ~~means~~indicates the associate nodes/threads which receive a common list of TCP/IP addresses for communication purposes; "queue" ~~means~~denotes a list of system resources (threads, memory, sockets, etc.); "quiesce" ~~mean to ensure~~signifies ensuring that all threads of an associated note have ceased processing problems; "head and tail pointers" ~~mean~~denote the link list of the list pointer (head pointer) and the link list end of list pointer (tail pointer); and "parallelization" ~~means~~denotes the conversion of a mathematical function or algorithm to its mathematical series equivalent. --